



Nicolas Vitale

# Optimising connectivity for algorithmic FX trading

Nicolas Vitale is a Director at Alpha Novae

If you're new to FX algorithmic trading, one of the first technological challenges you will face is connectivity. And it is not a trivial one. Despite what you might think, optimising connectivity is not something to be left only to high frequency trading firms, as in our current times, bad latency means money left on the table for each trade you make. You may already have a solid background in equities algorithmic trading, and you might consequently treat this question as a "déjà vu", but this would be a mistake.

## THE CHALLENGES

To start with the good news. FX data complexity is quite low when compared to other asset classes, and the volume of data is also much lower than what you can have on equities. For example, EBS Live, perceived as one of the main and fastest FX data feeds available

on the market, "only" publishes updates once every 100ms.

But in FX, which is by definition a decentralised market, you will have to deal with much more potential liquidity sources than what you would be used to for equities. You can trade currencies with banks, on their Single-Dealer or Multi-Dealer platforms, on an ever growing number of ECNs, on MTF, or even on Exchanges. To these different dealers and venues we should also add the latest players: an increasing number of HFT funds acting as "aggressive" liquidity providers. This new category is particularly booming since the SNB event earlier this year which has been followed by many historical Liquidity Providers leaving this business or seriously reducing their exposure and aggressiveness. Spreads have increased and if you want to lower them back, you will need the help of these HFT players.

A natural consequence of this decentralisation and diversity of participants, is the important geographical dispersion of the liquidity all over the world. So as opposed to equities, if you want to optimise your network latency, it is not as simple as collocating yourself in the datacentre of the matching engine of an Exchange. We can however simplify a bit the liquidity map and highlight three major geographical pools: London (LD4), New York (NY4) and Tokyo (TY3). It will then be up to you to calculate the optimal position of your servers depending where you will source your liquidity and what are the requirements of your algorithms.

Connecting to so many venues is a headache as most trading firms, especially on the buy-side, cannot invest massively in technology. So then comes the usual question, should you build or buy your connectivity?

## BUILD OR BUY

If you are in the high frequency and low latency trading business, technology is at the core of your alpha, and so you would certainly benefit by building your own connectivity infrastructure which you could then optimise and improve depending your specific requirements. By building in-house, you would even be able to optimise and tailor to your requirements the seven layers of the OSI model, from the physical layer (Ethernet, optical fibre or even microwaves) to the data-link, network or the transport layer (TCP, UDP, etc.) and the session, presentation or application layer (FIX, API, FAST, etc.).

Note that the two main layers generally looked at and optimised for our connectivity purpose are the "physical" and "application" layers.

Of course it would require an important investment to build, and to maintain over time an in-house infrastructure, but it is acceptable if it is to your competitive advantage, i.e. where you obtain the source of your alpha.

The buying option however makes sense for most buy-side firms. There is no massive up-front investment, no need for an in-house specialized team to support pro-actively the connection and its resiliency. Your time-to-market is also massively improved and you do not have the risk of project failure which always exists when building yourself such complex systems. Your time could be more efficiently spent on what matters to you really: the trading and the development of your models. Notice you would also still be able to select the physical or application layer you desire and consequently optimize the connectivity to your needs.

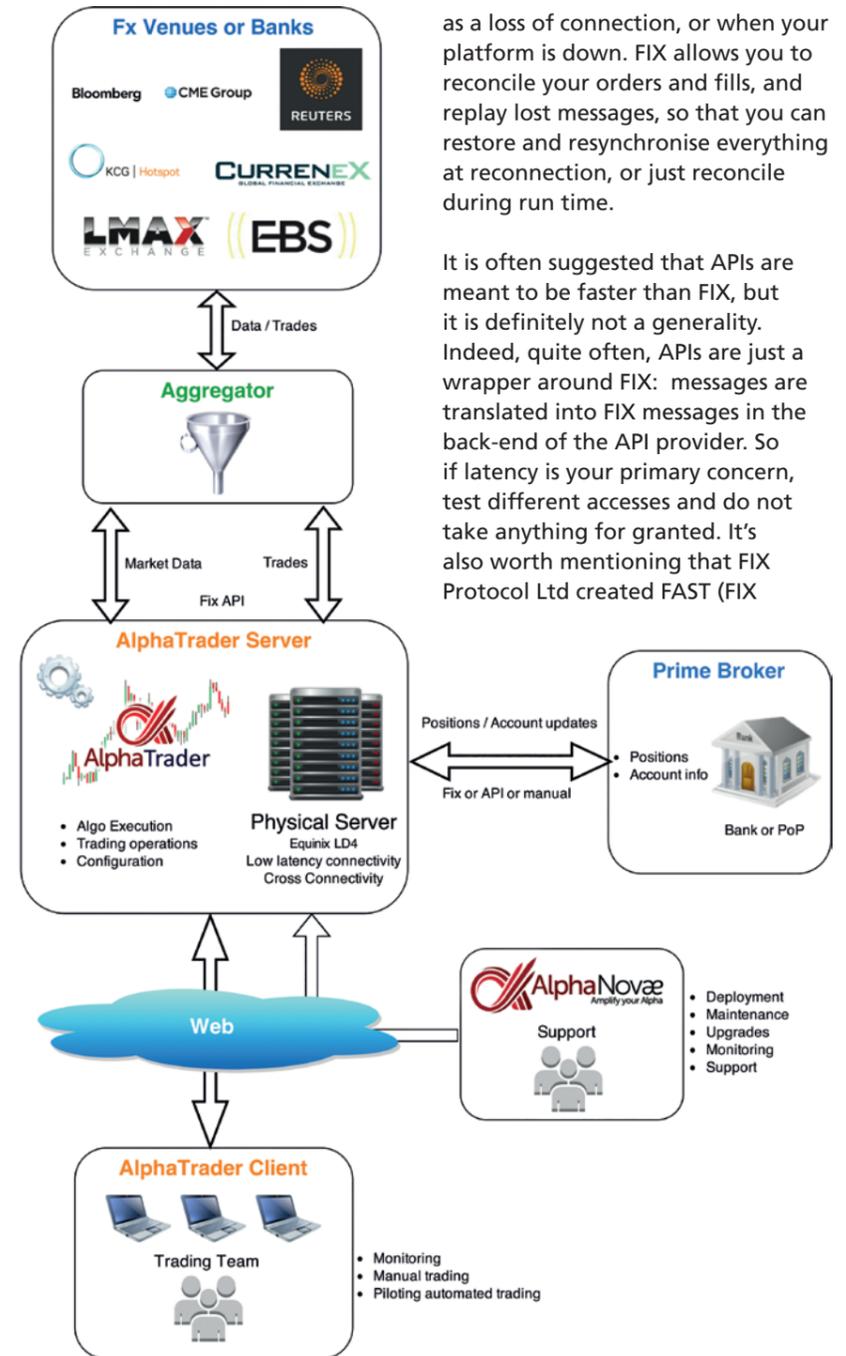
Bigger institutions, particularly on the sell-side, have historically been known to build their connectivity infrastructure in-house. But it's worth mentioning that in these times of very pressured budgets, they also tend to gradually shift their approach towards "buying". Building is not the default option anymore.

If you're not totally happy with either "build" or "buy", luckily, there is a third possible route. "Aggregators" can help you by bundling the access to the different banks and trading venues at the cost of only one single connection development and maintenance. You would still have to build and manage one connection to the aggregator, but they will be the one to manage on their back-end the aggregation of the data and routing of orders.

## FIX OR API

If you decide to build your own connectivity, you will quickly be facing the choice of the protocol to be used. By protocol, we mean here the one at the top of the OSI model, at the application layer, allowing your application to exchange information with the venue. Most of the time you will have the choice between either the standardised FIX protocol or a specific API built in-house.

The first obvious advantage associated to the FIX protocol, is that as a standard it facilitates the integration process. Once you have



as a loss of connection, or when your platform is down. FIX allows you to reconcile your orders and fills, and replay lost messages, so that you can restore and resynchronise everything at reconnection, or just reconcile during run time.

It is often suggested that APIs are meant to be faster than FIX, but it is definitely not a generality. Indeed, quite often, APIs are just a wrapper around FIX: messages are translated into FIX messages in the back-end of the API provider. So if latency is your primary concern, test different accesses and do not take anything for granted. It's also worth mentioning that FIX Protocol Ltd created FAST (FIX

built one FIX connection, adding others should be straightforward. Unfortunately reality is much more complex as different venues will implement different versions of FIX, and even for the same version, implementation may differ. You will also be surprised that generally only a few of the standardised FIX features will be implemented. In comparison with APIs, in term of functionalities it will look poor. To me, the main advantage of FIX is the resilience of the connection, especially in case of problems such

Adapted for Streaming) to support high-throughput and low latency data communications, but it is not commonly used in FX yet.

Because FIX implementations often lack useful features such as order commissions, aggregated account positions, account information and sometimes market data, it is often recommended or required to use APIs. That is why I believe that the best solution is to build hybrid connectors involving both FIX and API. You can then just pick the best from both.